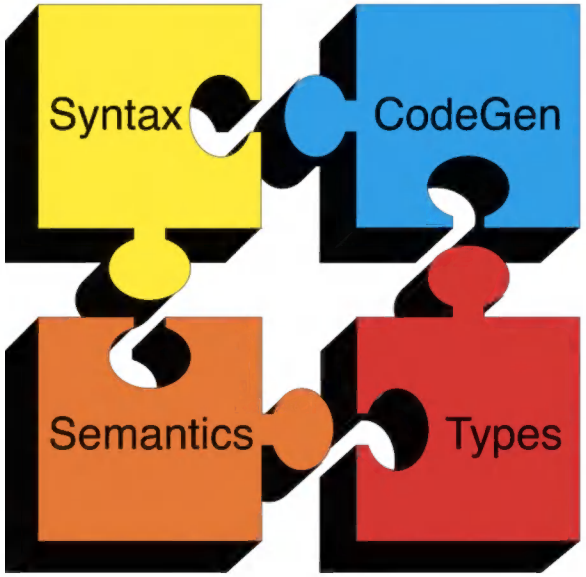

Compilers

Cool Semantics I

Alex Aiken

Cool Semantics I

$$\frac{}{so, E, S \vdash true : Bool(true), S}$$

$$\frac{}{so, E, S \vdash false : Bool(false), S}$$

$$\frac{i \text{ is an integer literal}}{so, E, S \vdash i : Int(i), S}$$

$$\frac{\begin{array}{c}s \text{ is a string literal} \\ n \text{ is the length of } s\end{array}}{so, E, S \vdash s : String(n,s), S}$$

Cool Semantics I

$$\frac{E(id) = l_{id} \qquad S(l_{id}) = v}{so, E, S \vdash id : v, S}$$

Cool Semantics I

$$\frac{}{so, E, S \vdash self : so, S}$$

Cool Semantics I

$$\frac{\begin{array}{c} so, E, S \vdash e : v, S_1 \\ E(id) = l_{id} \\ S_2 = S_1[v/l_{id}] \end{array}}{so, E, S \vdash id \leftarrow e : v, S_2}$$

Cool Semantics I

$$\frac{so, E, S \vdash e_1 : v_1, S_1 \quad so, E, S_1 \vdash e_2 : v_2, S_2}{so, E, S \vdash e_1 + e_2 : v_1 + v_2, S_2}$$

Cool Semantics I

$$\frac{so, E, S \vdash e_1 : v_1, S_1 \quad so, E, S_1 \vdash e_2 : v_2, S_2 \quad \ldots \quad so, E, S_{n-1} \vdash e_n : v_n , S_n}{so, E, S \vdash \{ e_1; \ldots e_n; \} : v_n, S_n}$$

Cool Semantics I

- Consider the expression
  - { X ← 7 + 5; 4; }

Cool Semantics I

$$\frac{so, E, S \vdash e_1 : Bool(true), S_1 \quad so, E, S_1 \vdash e_2 : v, S_2}{so, E, S \vdash \text{if } e_1 \text{ then } e_2 \text{ else } e_3 : v, S_2}$$

Cool Semantics I

$$\frac{so, E, S \vdash e_1 : Bool(false), S_1}{so, E, S \vdash \text{while } e_1 \text{ loop } e_2 \text{ pool} : void, S_1}$$

Cool Semantics I

$$\frac{so, E, S \vdash e_1 : Bool(true), S_1 \quad so, E, S_1 \vdash e_2 : v, S_2 \quad so, E, S_2 \vdash \text{while } e_1 \text{ loop } e_2 \text{ pool} : void, S_3}{so, E, S \vdash \text{while } e_1 \text{ loop } e_2 \text{ pool} : void, S_3}$$

Cool Semantics I

$$\frac{so, E, S \vdash e_1 : v_1, S_1 \quad so, ?, ? \vdash e_2 : v, S_2}{so, E, S \vdash \text{let id} : T \leftarrow e_1 \text{ in } e_2 : v_2, S_2}$$

- In what context should e_2 be evaluated?
  - Environment like E but with a new binding of id to a fresh location l_{new}
  - Store like S_1 but with l_{new} mapped to v_1

Cool Semantics I

- We write l_{new} = newloc(S) to say that l_{new} is a location not already used in S
  - newloc is like the memory allocation function

$$\frac{so, E, S \vdash e_1 : v_1, S_1 \quad l_{new} = newloc(S_1) \quad so, E[l_{new}/id] , S_1[v_1/l_{new}] \vdash e_2 : v_2, S_2}{so, E, S \vdash \text{let id} : T \leftarrow e_1 \text{ in } e_2 : v_2, S_2}$$

Cool Semantics I

Fill in the missing store value for the derivation of $(x \leftarrow 6) < x + 1$.

$$\dfrac{\dfrac{\text{so, } [x:l], S_1 \vdash 6 : \text{Int}(6), S_2 \quad S_3 = S_2[6/l]}{\text{so, } [x:l], S_1 \vdash x \leftarrow 6 : 6, S_3} \quad \dfrac{\text{so, } [x:l], S_3 \vdash 1 : \text{Int}(1), S_4 \quad \text{so, } [x:l], S_4 \vdash x : 6, S_5}{\text{so, } [x:l], S_3 \vdash x + 1 : 7, S_5}}{\text{so, } [x:l], [l \leftarrow 3] \vdash (x \leftarrow 6) < x + 1 : \text{Bool(true)}, S_5}$$

	S_2	S_3	S_4	S_5
○	$[l \leftarrow 3]$	$[l \leftarrow 3]$	$[l \leftarrow 6]$	$[l \leftarrow 7]$
○	$[l \leftarrow 6]$	$[l \leftarrow 6]$	$[l \leftarrow 7]$	$[l \leftarrow 7]$
○	$[l \leftarrow 3]$	$[l \leftarrow 3]$	$[l \leftarrow 6]$	$[l \leftarrow 6]$
○	$[l \leftarrow 3]$	$[l \leftarrow 6]$	$[l \leftarrow 6]$	$[l \leftarrow 6]$